ISSN 1342-2804

Research Reports on Mathematical and Computing Sciences

Sums of Squares Representation of Polynomials by Alternating Directional Augmented Lagrangian Methods with Fast Convergence

Hikaru Komeiji, Sunyoung Kim, Makoto Yamashita

March 2018, B–488

Department of Mathematical and Computing Sciences
Tokyo Institute of Technology

SERIES B: Operations Research

Sums of Squares Representation of Polynomials by Alternating Directional Augmented Lagrangian Methods with Fast Convergence

Hikaru Komeiji[*] Sunyoung Kim[†] Makoto Yamashita[‡]

March, 2018

Abstract

We study expressing a polynomial as sums of squares (SOS) of polynomials of lower degree by formulating the problem as semidefinite programs (SDPs). To efficiently solve the SDPs whose size grows very rapidly with the degree and number of variables of the polynomial, the alternative direction augmented Lagrangian (ADAL) method is investigated. We present conditions for the ADAL method to converge to an optimal solution in a few iterations and prove its convergence under the conditions. In addition, for the problem of representing a univariate trigonometric polynomial as an SOS, we also provide similar conditions for the fast convergence of the ADAL method to an optimal solution. Numerical results demonstrate the fast convergence of the method if the conditions are satisfied and the strictly feasible region is not very small. Test instances include SDPs with 11,476 variables and 22,533,126 constraints.

Key words. Sums of squares of polynomials, Sums of squares of univariate trigonometric polynomials, Semidefinite programs, Alternating directional augmented Lagrangian methods, Conditions for fast convergence.

AMS Classification. 90C22, 90C25, 90C26.

1 Introduction

We consider a problem of representing polynomials as sums of squares (SOS) of polynomials of lower degree using alternating direction augmented Lagrangian methods [16]. The

[*]Department of Mathematical and Computing Science, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552, Japan (tenryoki@gmail.com).

[†]Department of Mathematics, Ewha W. University, 52 Ewhayeodae-gil, Sudaemoon-gu, Seoul 03760, Korea (skim@ewha.ac.kr). The research was supported by NRF 2017-R1A2B2005119.

[‡]Department of Mathematical and Computing Science, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552, Japan (makoto.yamashita@c.titech.ac.jp). This research was partially supported by JSPS KAKENHI (Grant number: 15K00032).

problem arises from determining the nonnegativity of a polynomial $p(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathbb{R}^n$, which is known to be NP-hard for $n \geq 4$. As SOS polynomials are clearly nonnegative, representing a polynomial as SOS polynomials can be considered for the problem of testing the nonnegativity of a polynomial. Moreover, for a given nonnegative polynomial, finding an SOS representation of the polynomial is a central problem with many applications [2, 3, 11]. Thus, it is important to provide fast convergent algorithms for the problem.

The problem of representing a polynomial as an SOS can be formulated as a semidefinite program (SDP) [10]. For a polynomial $p(\boldsymbol{x})$ with degree d and n variables, the size of the resulting SDP becomes $\binom{n+d}{n}$. As n and d increase, the size of SDP grows very rapidly. SDPs are usually solved by SDP solvers such as SeDuMi [13], SDPA [17, 18], SDPT3 [14], and SDPNAL+ [19]. SDP solvers based on primal-dual interior-point algorithms, SeDuMi, SDPA and SDPT3, cannot efficiently solve SDP relaxations where the size of the variable matrix is more than several thousands if the structure and sparsity of polynomials are not exploited. SDPNAL+ [19] that employs a majorized semismooth Newton-CG augmented Lagrangian method and the second order information was shown to handle some large-scale SDPs.

The alternating direction augmented Lagrangian (ADAL) method, a first-order method, was proposed in [16] with the convergence result to efficiently solve SDPs. The ADAL method applies the alternating direction method to the dual augmented Lagrangian function. At each iteration, the dual augmented Lagrangian function is minimized with respect to three different variables in sequence: first the Lagrangian multipliers to the linear constraints, second the dual variables, and finally the primal variables. The ADAL method is different from other alternating direction methods where the variables are partitioned into several blocks and the augmented Lagrangian function is minimized over each block.

The main purpose of this paper is two folds: First, we present conditions for the ADAL method to converge in a few iterations to an optimal solution for the SOS representation problem of general polynomials. The proof for the fast convergence utilizes the sequential updating scheme of the variables in the ADAL method. Second, we also provide conditions for the fast convergence of the ADAL method to solve the SOS problem of univariate trigonometric polynomials, which have applications in filter design [5, 12]. In particular, we prove that a univariate trigonometric polynomial can be adjusted to satisfy the conditions by adding a positive number to the polynomial.

For the SOS representation problem of general polynomials, we show numerically that the conditions are more likely to be satisfied if a positive number is added to the polynomials. It is also numerically illustrated that the size of the strictly feasible region of the problem is closely related to whether a given problem satisfies the condition for the fast convergence of the ADAL method to an optimal solution. More precisely, the condition for the fast convergence presented in this paper is more likely to be satisfied by enlarging the strictly feasible region of the problem. We present numerical results in Section 5 that are consistent with the theoretical convergence results. The performance of the ADAL method is compared to that of SeDuMi [13]. The largest SDP instance includes 11,476 variables and 22,533,126 constraints. The number of constraints in this instance is two times larger than the largest problem solved by SDPNAL+ in [19]. We demonstrate that accurate optimal solutions are obtained much faster than SeDuMi, especially when the strictly feasible region is not very small and the conditions are satisfied. Moreover,

the ADAL method can solve much larger problems while SeDuMi terminates with out-of-memory error.

This paper is organized as follows: we briefly describe SDPs and the ADAL method in Section 2. In Section 3, we show the conditions for the fast convergence of the ADAL method and prove the fast convergence under the assumption that all the coefficient matrices of the SDP are linearly independent. Section 4 discusses the conditions for the SOS representation problem of trigonometric polynomials and prove the fast convergence of the ADAL method. In Section 5, numerical results are presented on sparse polynomials from [7], a dense polynomial and an illustrative example. Finally, we conclude in Section 6.

2 Preliminaries

2.1 Semidefinite programs

Let $\boldsymbol{e}$ be the vector of all ones and I the identity matrix in appropriate dimension. Let $\mathbb{R}^m$ denote m-dimensional Euclidian space and $\mathbb{S}^n$ the space of n-dimensional real symmetric matrices. The inner product of two symmetric matrices is defined as $X \bullet Y := \sum_{i,j} X_{ij} Y_{ij}$, $||X||_F := \sqrt{X \bullet X}$, and $X \succeq O$ means that X is positive semidefinite. We denote the space of n-dimensional positive semidefinite matrices by $\mathbb{S}^n_+$. The set of natural numbers and the set of integers are denoted by $\mathbb{N}$ and $\mathbb{Z}$, respectively.

The standard equality form of semidefinite programs (SDPs) can be expressed as

$$\begin{array}{lll} \text{(Primal)} \quad \min & C \bullet X & \\ \text{s.t.} & \mathcal{A}(X) = b & (1) \\ & \mathcal{S}^n \ni X \succeq O. & \end{array}$$

The dual is written as

$$\begin{array}{lll} \text{(Dual)} \quad \min & -b^T y & \\ \text{s.t.} & \mathcal{A}^*(y) + S = C & (2) \\ & \mathcal{S}^n \ni S \succeq O, & \end{array}$$

where $C, A_1, \ldots, A_m \in \mathcal{S}^n$ are data matrices.

The operator $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ is defined as $\mathcal{A}(X) := (A_1 \bullet X, A_2 \bullet X, \ldots, A_m \bullet X)$ and its adjoint operator $\mathcal{A}^* : \mathbb{R}^m \to \mathcal{S}^n$ is defined as $\mathcal{A}^*(y) := \sum_{i=1}^m A_i y_i$.

2.2 Alternating direction augmented Lagrangian methods

We describe that the ADAL method [16] for the primal-dual pair of SDP (1) and (2). For the primal SDP (1), the following assumption is necessary to ensure the convergence of the ADAL method.

Assumption 2.1. The coefficient matrices $\{A_i\}_{i=1}^m$ are linearly independent and there is an interior solution $\bar{X} \succ O$ for $\mathcal{A}(X) = b$.

Consider the augmented Lagrangian function defined with the penalty parameter $\mu > 0$:

$$\mathcal{L}_\mu(X, y, S) := -b^T y + X \bullet (\mathcal{A}^*(y) + S - C) + \frac{1}{2\mu}||\mathcal{A}^*(y) + S - C||_F^2.$$

The basic idea of the ADAL method is to minimize the augmented Lagrange function for the dual SDP (2) alternatively with respect to one of three types of variables: the Lagrangian multipliers y for the linear constraints, the dual variables S and the primal variables X:

$$\begin{aligned} y^{k+1} &:= \underset{y \in \mathbb{R}^m}{\operatorname{argmin}} \mathcal{L}_\mu(X_k, y, S_k) \\ S_{k+1} &:= \underset{S \in \mathcal{S}^n}{\operatorname{argmin}} \mathcal{L}_\mu(X_k, y^{k+1}, S) \\ X_{k+1} &:= X_k + \frac{1}{\mu}(\mathcal{A}^*(y^{k+1}) + S_{k+1} - C). \end{aligned}$$

These computations are included in the main steps of the ADAL algorithm described below:

Step 1. Initialize $X_0,\ S_0 \succeq O,\ \mu > 0$.

Step 2. At iteration $k+1$, $(k = 0, 1, ...)$

$$\begin{aligned} y^{k+1} &= -(\mathcal{A}\mathcal{A}^*)^{-1}\{\mu(\mathcal{A}(X_k) - b) + \mathcal{A}(S_k - C)\}, && (3) \\ W_{k+1} &= C - \mathcal{A}^*(y^{k+1}) - \mu X_k, && (4) \\ S_{k+1} &= \Pi_{\mathbb{S}^n_+}(W_{k+1}), && (5) \\ X_{k+1} &= \frac{1}{\mu}(S_{k+1} - W_{k+1}). && (6) \end{aligned}$$

Here, $\Pi_{\mathbb{S}^n_+}(W_{k+1})$ is the orthogonal projection of W_{k+1} onto $\mathbb{S}^n_+$.

For a termination criterion in the ADAL algorithm, we use the following:

$$\text{p\_inf} := \frac{||\mathcal{A}(X) - b||_2}{1 + ||b||_2},\ \text{d\_inf} := \frac{||\mathcal{A}^*(y) + S - C||_F}{1 + ||C||_F},\ \text{gap} := \frac{||b^T y - C \bullet X||}{1 + ||b^T y|| + ||C \bullet X||}. \quad (7)$$

With a threshold value "tol", the ADAL algorithm is terminated if

$$\max\{\text{p\_inf, d\_inf, gap}\} \leq \text{tol}. \quad (8)$$

The following lemma is included for the theoretical convergence analysis of the ADAL method.

Lemma 2.2. *If Assumption 2.1 is satisfied, there exist points that satisfy the KKT condition.*

$$\mathcal{A}(X) = b,\ \mathcal{A}^*(y) + S = C,\ SX = 0,\ X \succeq O,\ S \succeq O.$$

By using the theory of the fixed points, it is shown that the ADAL method converges from an arbitrary initial point to the KKT point in Lemma 2.2. See Section 2.2 in [16] for details.

2.3 Sums of squares representations

We discuss representing nonnegative polynomials as sums-of-squares of polynomials. Let $\mathbb{R}[\boldsymbol{x}]$ be a n variable polynomial ring where $\boldsymbol{x} := (x_1, x_2, \ldots, x_n)$. We also let $\mathbb{Z}_+$ denote the set of nonnegative integers. For the support verctor $\boldsymbol{\alpha} := (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{Z}_+^n$, a monomial is denoted by $\boldsymbol{x}^{\boldsymbol{\alpha}} := x_1^{\alpha_1} x_2^{\alpha_2} ... x_n^{\alpha_n}$, a coefficient of $\boldsymbol{x}^{\boldsymbol{\alpha}}$ in $p(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ by $c_{\boldsymbol{\alpha},p} \in \mathbb{R}$, and the set of supports of monomials with nonzero coefficients by $\mathcal{F}_p := \{\boldsymbol{\alpha} \in \mathbb{Z}_+^n | c_{\boldsymbol{\alpha},p} \neq 0\}$. Then $p(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ is expressed as $p(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{F}_p} c_{\boldsymbol{\alpha},p} \boldsymbol{x}^{\boldsymbol{\alpha}}$. We denote the degree of p by $\deg(p) := \max\{\sum_{i=1}^n \alpha_i | \boldsymbol{\alpha} \in \mathcal{F}_p\}$.

A polynomial $p(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ is an SOS if

$$\exists\ q_i(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]\ (i = 1, 2, \ldots, m) \text{ such that } p(\boldsymbol{x}) = \sum_{i=1}^m q_i^2(\boldsymbol{x}).$$

Let $\mathcal{P}_n = \{p(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}] \mid p(\boldsymbol{x}) \geq 0\ (\forall \boldsymbol{x} \in \mathbb{R}^n)\}$ denote the set of nonnegative polynomials and $\Sigma_n = \{p(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}] \mid p(\boldsymbol{x}) \text{ is SOS}\}$ the set of SOS polynomials. Using the following lemma, representing a polynomial as an SOS can be formulated as an SDP.

Lemma 2.3. [10] *A polynomial $p(\boldsymbol{x}) \in \mathbb{R}[\boldsymbol{x}]$ is in Σ_n if and only if there exists $X \in \mathcal{S}^N$ such that $p(\boldsymbol{x}) = u_d(\boldsymbol{x})^T X u_d(\boldsymbol{x})$, and $X \succeq O$, where $u_d(\boldsymbol{x}) = (1, x_1, \ldots, x_n, x_1^2, x_1 x_2, \ldots, x_n^d)$, $d \geq \lceil deg(p)/2 \rceil$ and $N = \binom{n+d}{n}$.*

To represent $p(\boldsymbol{x})$ as an SOS, we consider the following problem called the SOS feasibility problem:

$$\text{Find } X \text{ such that } \ p(\boldsymbol{x}) = u_d(\boldsymbol{x})^T X u_d(\boldsymbol{x}),\ X \succeq O. \tag{9}$$

2.4 Formulating SOS feasibility problems as SDPs

The SOS feasibility problem (9) can be formulated as the equality standard form SDP (1) in two different ways using the coefficients of the functions [4] and the function values [9].

First, we describe the SDP formulation by equating the coefficients of the functions. Let $\mathcal{G}_{2d} := \{\boldsymbol{\alpha} \in \mathbb{Z}_+^n | \sum_{i=1}^n a_i \leq 2d\}$ denote the set of supports of terms whose degree is not greater than $2d$. We denote $E_{\boldsymbol{\alpha}}$ as a constant matrix whose elements for $\boldsymbol{x}^{\boldsymbol{\alpha}}$ in $u_d(\boldsymbol{x}) u_d(\boldsymbol{x})^T$ are 1's and zeros elsewhere. Then, the constraint of the SOS feasibility problem (9) can be equivalently expressed as

$$\sum_{\boldsymbol{\alpha} \in \mathcal{F}_p} c_{\boldsymbol{\alpha},p} \boldsymbol{x}^{\boldsymbol{\alpha}} = u_d(\boldsymbol{x})^T X u_d(\boldsymbol{x}),\ X \succeq O \ \Leftrightarrow \sum_{\boldsymbol{\alpha} \in \mathcal{G}_{2d}} c_{\boldsymbol{\alpha},p} \boldsymbol{x}^{\boldsymbol{\alpha}} = X \bullet u_d(\boldsymbol{x}) u_d(\boldsymbol{x})^T,\ X \succeq O,$$

$$\Leftrightarrow \quad \sum_{\boldsymbol{\alpha} \in \mathcal{G}_{2d}} c_{\boldsymbol{\alpha},p} \boldsymbol{x}^{\boldsymbol{\alpha}} = X \bullet \sum_{\boldsymbol{\alpha} \in \mathcal{G}_{2d}} E_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\alpha} \in \mathcal{G}_{2d}} X \bullet E_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}},\ X \succeq O.$$

Thus, the following constraints are obtained:

$$c_{\boldsymbol{\alpha},p} = X \bullet\ E_{\boldsymbol{\alpha}}\ (\boldsymbol{\alpha} \in \mathcal{G}_{2d}).$$

The SOS feasibility problem (9) is formulated as an SDP with an appropriately chosen objective function:

$$\begin{array}{lll} \text{(Primal)} & \min & \mathrm{Diag}(1,O) \bullet \mathrm{Diag}(\hat{x},X) \\ & \text{s.t.} & \mathrm{Diag}(0,E_{\boldsymbol{\alpha}}) \bullet \mathrm{Diag}(\hat{x},X) = c_{\boldsymbol{\alpha},p} \ (\boldsymbol{\alpha} \in \mathcal{G}_{2d}), \\ & & \hat{x} \geq 0, Q \succeq O. \\ \text{(Dual)} & \min & -\sum_{\boldsymbol{\alpha}\in\mathcal{G}_{2d}} c_{\boldsymbol{\alpha},p}\tilde{y}_{\boldsymbol{\alpha}} \\ & \text{s.t.} & \sum_{\boldsymbol{\alpha}\in\mathcal{G}_{2d}} \mathrm{Diag}(0,E_{\boldsymbol{\alpha}})\tilde{y}_{\boldsymbol{\alpha}} + \mathrm{Diag}(\hat{s},S) = \mathrm{Diag}(1,O), \\ & & \hat{s} \geq 0, S \succeq O, \end{array} \tag{10}$$

where $N = \binom{n+d}{n}$, $X, S \in \mathbb{S}^N$ and $\mathrm{Diag}(\hat{x}, X) \in \mathbb{S}^{N+1}$ means the block-diagonal matrix whose upper leftmost element is $\hat{x}$ and $N \times N$ lower-right submatrix is X. The size of the vector $[\tilde{y}_{\boldsymbol{\alpha}}]$ $(\boldsymbol{\alpha} \in \mathcal{G}_{2d})$ is $\binom{n+2d}{n}$. As a result, it becomes very hard to solve the SDP problem as n and d increase.

Problem (10) can be expressed in the form of SDP (1) and (2). We use the following definition:

$$\begin{array}{l} A_{\boldsymbol{\alpha}} := E_{\boldsymbol{\alpha}}, \ \tilde{A}_{\boldsymbol{\alpha}} := \mathrm{Diag}(0, E_{\boldsymbol{\alpha}}), \ \tilde{b}_{\boldsymbol{\alpha}} := c_{\boldsymbol{\alpha},p}, \ \tilde{C} := \mathrm{Diag}(1, O), \\ \text{and } \tilde{X} := \mathrm{Diag}(\hat{x}, X), \ \tilde{S} := \mathrm{Diag}(\hat{s}, S). \end{array} \tag{11}$$

If $\mathcal{A}, b, C, X, y, S$ are replaced with $\tilde{\mathcal{A}}, \tilde{b}, \tilde{C}, \tilde{X}, \tilde{y}, \tilde{S}$, then the SDP (10) can be written in the form of (1) and (2). We note that the resulting SDP has the following structure.

Remark 2.4. (The structure of the SDP constructed by equating coefficients)

(i) $\tilde{\mathcal{A}}(\tilde{C}) = \mathbf{0}$.

(ii) All coefficient matrices are block diagonal.

Next, we formulate the SOS feasibility problem as the standard equality form SDP (1) by deriving constraints based on the function values [9]. In particular, we focus on the univariate trigonometric polynomial. SOS representations of trigonometric polynomials have been widely used in filter design [5, 12]. We discuss the SOS feasibility problem of trigonometric polynomials in Section 4.

Definition 2.5. The univariate trigonometric polynomial of degree N is defined as

$$p_N(t) = p_0 + \sum_{k=1}^{N} (p_k \cos(kt) + p_{-k} \sin(kt)).$$

As $p_N(t)$ has $2N + 1$ coefficients $(p_{-N}, p_{-N+1}, \ldots, p_{-1}, p_0, p_1, \ldots, p_{N-1}, p_N)$, we take $2n+1$ sampling points $t_1, t_2, \ldots, t_{2N+1}$ and derive an SDP by equating the function values at the sample points. More specifically, we use the following lemma to construct an SDP to represent p_N as an SOS.

Lemma 2.6. ([9]) *The univariate trigonometric polynomial $p_N(t)$ of degree N is nonnegative if and only if there exists a matrix $X \in \mathbb{S}_+^{N+1}$ such that $p_N(t) = v(t)^T X v(t)$. Here, the vector $v(t)$ is defined by*

$$v(t) := \left(\cos\left(\frac{t}{2}\right), \sin\left(\frac{t}{2}\right), \ldots, \cos\left(k+\frac{1}{2}\right)t, \sin\left(k+\frac{1}{2}\right)t\right)^T$$

for $N = 2k+1$ $(k \in \mathbb{N})$, and

$$v(t) := (1, \cos(t), \sin(t), \ldots, \cos(kt), \sin(kt))^T$$

for $N = 2k$ $(k \in \mathbb{N})$.

By Lemma 2.6, the SOS feasibility problem can be written as

$$\text{Find } X \text{ such that } \ X \succeq O,\ p_N(t_i) = v(t_i)^T X v(t_i)\ (i = 1, 2, \ldots, 2N+1). \tag{12}$$

With an appropriate objective function, we obtain the following SDP:

$$\begin{array}{lll}
\text{(Primal)} & \min & \mathrm{Diag}(1, O) \bullet \mathrm{Diag}(\hat{x}, X) \\
& \text{s.t.} & \mathrm{Diag}(0, v(t_i)v(t_i)^T) \bullet \mathrm{Diag}(\hat{x}, X) = p_N(t_i)\ (i = 1, 2, \ldots, 2N+1), \\
& & \hat{x} \geq 0,\ \mathbb{S}^{N+1} \ni X \succeq O. \\
\text{(Dual)} & \min & -\sum_{i=1}^{2N+1} p_N(t_i)\tilde{y}_i \\
& \text{s.t.} & \sum_{i=1}^{2N+1} \mathrm{Diag}(0, v(t_i)v(t_i)^T)\tilde{y}_i + \mathrm{Diag}(\hat{s}, S) = \mathrm{Diag}(1, O), \\
& & \hat{s} \geq 0,\ \mathbb{S}^{N+1} \ni S \succeq O.
\end{array} \tag{13}$$

Problem (13) can also be written as the equality standard form SDP (1) and (2) by letting

$$\begin{array}{l}
A_i := v(t_i)v(t_i)^T,\ \tilde{A}_i := \mathrm{Diag}(0, v(t_i)v(t_i)^T),\ \tilde{b}_i := p_N(t_i),\ \tilde{C} := \mathrm{Diag}(1, O), \\
\text{and } \tilde{X} := \mathrm{Diag}(\hat{x}, X),\ \tilde{S} := \mathrm{Diag}(\hat{s}, S).
\end{array} \tag{14}$$

The resulting SDP also has the structure described in Remark 2.4. A difference from (10) is that the rank of all the coefficient matrices is 1, which reduces the computational complexity for the matrix-vector product. For this reason, SDP solver DSDP[1] can compute search directions for the primal-dual interior point method more efficiently than other solvers such as SeDuMi, SDPT3 and SDPA [13, 14, 17, 18].

We mention that it is difficult to choose sampling points in general. Good sampling methods for multivariate polynomials are not known. It is indicated in [9] that the best sample points for the univariate trigonometric polynomials in Definition 2.5 are taken in $[-\pi, \pi]$ as

$$t_i = -\pi + \frac{i-1}{2N+1}2\pi\ (i = 1, 2, ..., 2N+1). \tag{15}$$

3 Fast convergence

In this section, we discuss our fast convergence theorem for the ADAL method to solve the SOS feasibility problems (10) and (13). If the ADAL method satisfies some conditions

during the iteration, then it converges to an optimal solution that satisfies the KKT conditions in Lemma 2.2. In fact, the convergence to an optimal solution is guaranteed within a few iterations after satisfying the conditions. We call the convergence as *the fast convergence* as opposed to the tail convergence that is generally found in first-order methods.

When the ADAL method is used to solve the SOS feasible problems (10) and (13), the update in (4) corresponds to

$$\tilde{W}_{k+1} = \tilde{C} - \tilde{\mathcal{A}}^*(\tilde{y}^{k+1}) - \mu\tilde{X}_k.$$

Let $\tilde{W}_{k+1} = \mathrm{Diag}(w_{k+1}, W_{k+1})$ be the block-diagonal expression of $\tilde{W}_{k+1}$. The following theorem provides the conditions for the fast convergence.

Theorem 3.1. (Conditions for the fast convergence)
If $\tilde{W}_{k+1}$ satisfies the following conditions, an exact optimal solution is obtained at iteration $k+3$:

$$w_{k+1} \geq 0,\ W_{k+1} \preceq O,\ \textit{and}\ W_{k+1} \preceq \mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(S_k)\big).$$

Proof. From (11), (14) and Remark 2.4, we have

$$\tilde{\mathcal{A}}\tilde{\mathcal{A}}^* = \mathcal{A}\mathcal{A}^*,\ \tilde{\mathcal{A}}(\tilde{C}) = 0,\ \tilde{\mathcal{A}}(\tilde{X}) = \mathcal{A}(X),\ \text{and}\ \tilde{\mathcal{A}}(\tilde{S}) = \mathcal{A}(S).$$

Thus, $\tilde{y}$ is updated by (3) as

$$\begin{aligned}\tilde{y}^{k+1} &= -(\tilde{\mathcal{A}}\tilde{\mathcal{A}}^*)^{-1}\big\{\mu(\tilde{\mathcal{A}}(\tilde{X}_k) - \tilde{b}) + \tilde{\mathcal{A}}(\tilde{S}_k - \tilde{C})\big\} \\ &= -(\mathcal{A}\mathcal{A}^*)^{-1}\big\{\mu(\mathcal{A}(X_k) - \tilde{b}) + \mathcal{A}(S_k)\big\}. \end{aligned} \tag{16}$$

In (4), $\tilde{W}$ is computed by $\tilde{W}_{k+1} = \tilde{C} - \tilde{\mathcal{A}}^*(y^{k+1}) - \mu\tilde{X}_k$, thus each block can be obtained using the structure of $\tilde{A}_i$ and $\tilde{C}$ in Remark 2.4 as

$$\begin{aligned} w_{k+1} &= 1 - \mu\hat{x}_k \\ W_{k+1} &= -\mathcal{A}^*(\tilde{y}^{k+1}) - \mu X_k. \end{aligned} \tag{17}$$

From the assumptions $w_{k+1} \geq 0$ and $W_{k+1} \preceq O$, the matrix variables $\tilde{X}$ and $\tilde{S}$ are updated by (5) and (6) as follows:

$$\hat{x}_{k+1} = 0,\ X_{k+1} = -\frac{1}{\mu}W_{k+1} \quad \text{and} \quad \hat{s}_{k+1} = 1 - \mu\hat{x}_k,\ S_{k+1} = O.$$

From (3), (16) and (17), we further obtain

$$\begin{aligned}\tilde{y}^{k+2} &= -(\tilde{\mathcal{A}}\tilde{\mathcal{A}}^*)^{-1}\big\{\mu(\tilde{\mathcal{A}}(\tilde{X}_{k+1}) - \tilde{b}) + \tilde{\mathcal{A}}(\tilde{S}_{k+1} - \tilde{C})\big\} \\ &= -(\mathcal{A}\mathcal{A}^*)^{-1}\big\{\mu(\mathcal{A}(X_{k+1}) - \tilde{b}) + \mathcal{A}(S_{k+1})\big\} \\ &= -(\mathcal{A}\mathcal{A}^*)^{-1}\big\{\mu(\mathcal{A}(-W_{k+1}/\mu) - \tilde{b})\big\} \\ &= -(\mathcal{A}\mathcal{A}^*)^{-1}\big\{\mu(\mathcal{A}(X_k) + \mathcal{A}\mathcal{A}^*(\tilde{y}^{k+1})/\mu - \tilde{b})\big\} \\ &= -\tilde{y}^{k+1} - (\mathcal{A}\mathcal{A}^*)^{-1}\big\{\mu(\mathcal{A}(X_k) - \tilde{b}) + \mathcal{A}(S_k) - \mathcal{A}(S_k)\big\} \\ &= (\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(S_k), \end{aligned} \tag{18}$$

$$\begin{aligned} w_{k+2} &= 1 - \mu\hat{x}_{k+1} = 1, \\ W_{k+2} &= -\mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(S_k)\big) - \mu X_{k+1} \\ &= -\mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(S_k)\big) + W_{k+1}. \end{aligned} \tag{19}$$

From the assumption $W_{k+1} \preceq \mathcal{A}^*(\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(S_k)$, we have $W_{k+2} \preceq O$. Hence, we obtain

$$\hat{x}_{k+2} = 0,\ X_{k+2} = -\frac{1}{\mu}W_{k+2} \text{ and } \hat{s}_{k+2} = 1,\ S_{k+2} = O.$$

At iteration $k+3$,

$$\begin{aligned} \tilde{y}^{k+3} &= (\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(S_{k+1}) = 0, \\ w_{k+3} &= w_{k+2} = 1, \\ W_{k+3} &= W_{k+2} \preceq O. \end{aligned} \tag{20}$$

Finally, we obtain

$$\hat{x}_{k+3} = 0,\ X_{k+3} = -\frac{1}{\mu}W_{k+3} = -\frac{1}{\mu}W_{k+2} \quad \text{and} \quad \hat{s}_{k+3} = 1,\ S_{k+3} = O. \tag{21}$$

Next, we show that the solution $(\tilde{X}_{k+3}, \tilde{y}^{k+3}, \tilde{S}_{k+3})$ satisfies the KKT condition in Lemma 2.2. Using (19), (17) and (16),

$$\begin{aligned} X_{k+3} &= -\frac{1}{\mu}W_{k+2} = -\frac{1}{\mu}W_{k+1} + \frac{1}{\mu}\mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(S_k)\big) \\ &= X_k + \frac{1}{\mu}\mathcal{A}^*(\tilde{y}^{k+1}) + \frac{1}{\mu}\mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(S_k)\big) = X_k - \mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}(X_k) - \tilde{b})\big). \end{aligned}$$

Thus, the primal constraint is

$$\tilde{\mathcal{A}}(\tilde{X}_{k+3}) = \mathcal{A}(X_{k+3}) = \mathcal{A}\bigg(X_k - \mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}(X_k) - \tilde{b})\big)\bigg) = \tilde{b},$$

and the dual constraint is also satisfied by (20) and (21):

$$\tilde{\mathcal{A}}^*(\tilde{y}^{k+3}) + \tilde{S}_{k+3} = \mathrm{Diag}(0+1, \mathcal{A}^*(0) + O) = \mathrm{Diag}(1, O).$$

For the duality gap,

$$\tilde{X}_{k+3} \cdot \tilde{S}_{k+3} = 0 \times 1 + X_{k+3} \cdot O = 0.$$

Consequently, we have confirmed that an optimal solution satisfying Lemma 2.2 is obtained. □

Using Theorem 3.1, we show that an optimal solution can be obtained in two iterations under some condition.

Corollary 3.2. (Convergence in two iterations) *Take the initial points* $\hat{x}_0 \leq \frac{1}{\mu}, \tilde{S}_0 = \mathrm{Diag}(\hat{s}_0, O), \hat{s}_0 \in \mathbb{R}$. *If, for some* X_0,

$$X_0 - \mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}(X_0) - \tilde{b})\big) \succeq O$$

holds, then an exact solution satisfying the KKT condition of Lemma 2.2 is obtained in two iterations.

Proof. At the first iteration, from (16) and (17), we have

$$\begin{aligned} \tilde{y}^1 &= -(\mathcal{A}\mathcal{A}^*)^{-1}\{\mu(\mathcal{A}(X_0)-\tilde{b})\}, \\ w_1 &= 1-\mu\hat{x}_0 \geq 0, \\ W_1 &= -\mathcal{A}^*(\tilde{y}^1)-\mu X_0 = -\mu\{X_0-\mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}(X_0)-\tilde{b})\big)\} \preceq O. \end{aligned}$$

Hence, we have

$$\hat{x}_1 = 0,\ X_1 = -\frac{1}{\mu}W_1, \text{ and } \hat{s}_1 = 1-\mu\hat{x}_0,\ S_1 = O.$$

For the second iteration, by (18),

$$\begin{aligned} \tilde{y}^2 &= (\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(S_1) = 0, \\ w_2 &= 1, \\ W_2 &= -\mathcal{A}^*(\tilde{y}^2)-\mu X_1 = W_1. \end{aligned}$$

Thus, we obtain

$$\hat{x}_2 = 0,\ X_2 = -\frac{1}{\mu}W_2 = -\frac{1}{\mu}W_1 = X_0-\mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}(X_0)-\tilde{b})\big), \text{ and } \hat{s}_2 = 1,\ S_2 = O.$$

Similarly to the proof for Theorem 3.1, we can show that this solution satisfies the KKT condition. □

4 Sums of squares of trigonometric polynomials

In this section, we are concerned with the trigonometric polynomial. We prove that the fast convergence can be obtained as the minimum value of the trigonometric polynomial p_N becomes larger than 0. We also show that the strictly feasible region is large in such SOS feasibility problems. In addition, we describe the range where the conditions for convergence in two iterations hold with the same initial point.

4.1 Dirichlet kernel

For SOS representations of trigonometric polynomials, we introduce the Dirichlet kernel $K_N(t)$.

Definition 4.1. (The Dirichlet kernel) The Dirichlet kernel for $N \in \mathbb{N}$ is defined by

$$K_N(t) = \frac{1}{N}\frac{\sin\frac{Nt}{2}}{\sin\frac{N}{2}}.$$

For the odd and even cases, the kernel is equivalent to

$$\begin{cases} K_N(t) = \frac{1}{N}\sum_{k=-m}^{m}\cos kt & (N=2m+1, m\in\mathbb{N}) \\ K_N(t) = \frac{2}{N}\sum_{k=0}^{m}\cos(k+\frac{1}{2})t & (N=2m+2, m\in\mathbb{N}). \end{cases}$$

It is known that the following holds for the Dirichlet kernel:

$$\sum_{k=0}^{N-1} K_N^2(t-kw) \;\; = \;\; 1 \; (w := 2\pi/N). \tag{22}$$

We use the following lemma to represent the univariate nonnegative trigonometric polynomial of degree N as an SOS.

Lemma 4.2. ([9]) *If the univariate trigonometric polynomial of degree N is nonnegative, there exists $X \in \mathbb{S}_+^{N+1}$ such that $p_N(t) = v(t)^T X v(t)$ where $v(t) := \big(K_{N+1}(t), K_{N+1}(t-\tau), \ldots, K_{N+1}(t-N\tau)\big)^T$ with $\tau = \frac{2\pi}{N+1}$.*

The sampling points $t_1, t_2, \ldots, t_{2N+1}$ are equally spaced nodes in $[-\pi, \pi]$ and the number of points is determined by the number of the coefficients of $p_N(t)$ as (15).

4.2 Conditions for fast convergence

We consider the conditions for fast convergence discussed in Section 3 for SOS representations of trigonometric polynomials with the Dirichlet kernel. We first discuss an important property for the SOS feasibility problem of trigonometric polynomials (13).

Lemma 4.3. *When representing a univariate trigonometric polynomial as an SOS using the Dirichlet kernel,* (13) *satisfies*

$$\mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}\boldsymbol{e}\big) = I \text{ and } \mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(I)\big) = I.$$

Proof. See Appendix A. □

Using Lemma 4.3, we have the following condition for fast convergence.

Corollary 4.4. *Let the initial points be $\tilde{X}_0 = \mathrm{Diag}(\hat{x}_0, \beta I)$ $(\hat{x}_0 \leq \frac{1}{\mu}, \beta \in \mathbb{R}), \tilde{S}_0 = \mathrm{Diag}(0, O)$, the condition for convergence in two iterations for* (13) *is*

$$\mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}\tilde{b}\big) \succeq O.$$

Proof. The condition for convergence in two iterations from Corollary 3.2 is

$$X_0 - \mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}(X_0) - \tilde{b})\big) \succeq O.$$

From Lemma 4.3, $\mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(X_0)\big) = \beta I$. Thus, with the initial points $\tilde{X}_0 = \mathrm{Diag}(\hat{x}_0, \beta I)$ $(\hat{x}_0 \leq \frac{1}{\mu}, \beta \in \mathbb{R}), \tilde{S}_0 = \mathrm{Diag}(0, O)$, the desired result follows. □

4.3 Convergence conditions for minimizing $p_N(t)$

We discuss how to adjust a given $p_N(t)$ to satisfy the condition for convergence in two iterations in Corollary 4.4. First, define $p_{N,\gamma}(t) := p_N(t) + \gamma$ ($\gamma \in \mathbb{R}$) and consider minimizing $p_{N,\gamma}(t)$ in the form of SDP (13) by taking the coefficient matrices as (14) and replacing $\tilde{b}_i$ by $\tilde{b}_i + \gamma := p_{N,\gamma}(t_i)$. Thus, the condition for convergence in two iterations in Corollary 3.2 for $p_{N,\gamma}(t)$ becomes

$$\begin{aligned} & X_0 - \mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}(X_0) - \tilde{b} - \gamma e)\big) \succeq O, \\ \text{or} \quad & X_0 - \mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}(X_0) - \tilde{b})\big) + \gamma I \succeq O \quad \text{(using Lemma 4.3).} \end{aligned} \tag{23}$$

As γ becomes large, $p_{N,\gamma}$ is more likely to satisfy the convergence condition described above in view of Corollary 3.2. In other words, the condition for convergence is more likely satisfied for p_N when the minimum value of p_N is larger than 0.

We now discuss on the value of γ in $p_{N,\gamma}$ to satisfy the condition for convergence in two iterations. Let $\breve{X} = X_0 - \mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}(X_0) - \tilde{b})\big)$ and $\lambda_{\min}(\breve{X})$ be the minimum eigenvalue of $\breve{X}$. Then, (23) can be written as

$$\breve{X} + \gamma I \succeq O.$$

For the SOS feasibility problem (13) of p_N, the condition for convergence in two iteration is $\breve{X} \succeq O$. If $\breve{X} \prec O$, then $\breve{X} + \gamma I \succeq O$ holds when $\gamma \geq -\lambda_{\min}(\breve{X})$. Thus, the condition for convergence in two iterations for $p_{N,\gamma}$ is satisfied if $\gamma \geq -\lambda_{\min}(\breve{X})$.

Lemma 4.5. *For the SOS feasibility problem (13) of p_N, assume that the initial points $\breve{X}_0 = Diag(\hat{x}_0, X_0), \hat{x}_0 \leq \frac{1}{\mu}, \tilde{S}_0$ and Corollary 3.2 holds. Then, if $\gamma \geq -\lambda_{\min}(\breve{X})$, the condition for convergence in two iterations is satisfied for $p_{N,\gamma}$.*

4.4 The size of the strictly feasible region of the SOS feasibility problem of trigonometric polynomials

We show that finding the minimum value of a nonnegative trigonometric polynomial is equivalent to computing the size of the strictly feasible region of the SOS feasibility problem (24) using the Dirichlet kernel.

The size of the strictly feasible region of the SOS feasibility problems (10) and (13) can be measured by solving an SDP. More precisely, we show that the following SDP is solved for the size of the strictly feasible region:

$$\max_{\delta \in \mathbb{R}} \ \delta \text{ subject to } \tilde{\mathcal{A}}(\delta \tilde{I} + \tilde{X}) = \tilde{b}, \ \tilde{X} \succeq O, \ \delta \geq 0, \tag{24}$$

where $\tilde{I}$ is an identity matrix of the same size as $\tilde{X}$. If δ is equal to 0, it means that no interior solution exists, thus Assumption 2.1 does not hold. If it is positive, interior solutions exist and the optimal value δ represents the size of the strictly feasible region of (10) and (13).

Using (22), we know $A_i \bullet I = v(t_i)v(t_i)^T \bullet I = v(t_i)^T v(t_i) = 1$ $(i = 1, 2, \ldots, 2N+1)$. From $\tilde{\mathcal{A}}(\tilde{X}) = \mathcal{A}(X)$, (24) can be expressed as

$$\max_{\delta \in \mathbb{R}} \ \delta \quad \text{subject to } \delta \boldsymbol{e} + \mathcal{A}(X) = \tilde{b}, \ X \succeq O, \ \delta \geq 0. \tag{25}$$

With $\tilde{b}_i = p_N(t_i)(i = 1, 2, ..., 2N+1)$, (25) can be equivalently written as

$$\max_{\delta \in \mathbb{R}} \ \delta \text{ subject to } \ p_N(t_i) - \delta = v(t_i)^T X v(t_i) \ (i = 1, 2, \ldots, 2N+1), \ X \succeq O, \ \delta \geq 0. \tag{26}$$

From (12), (26) is equivalent to a minimization problem for nonnegative p_N. As a result, the strictly feasible region of the trigonometric polynomial that satisfies the condition for convergence in two iterations is obtained by maximizing δ. Note that (26) does not have the structure mentioned in Remark 2.4 (i).

5 Numerical experiments

We present numerical results on the ADAL algorithm. The purposes of our experiments are

- to verify the fast convergence for the SOS feasibility problems of general and trigonometic polynomials.
- to demonstrate that the ADAL method performs better than SeDuMi which is based on the primal-dual interior point method.
- to compare the size of the strictly feasible region of the polynomials that converge rapidly with that of polynomials with slow convergence for the SOS feasibility problem.
- to show that the performance of the ADAL can be different depending on the formulations of the SOS feasibility problem for a given problem.

Our numerical experiments were conducted on a Mac with 2.7 GHz Intel Core i5, 16 GB memory space. The ADAL algorithm was implemented in MATLAB R2017b.

All test SOS feasibility problems are formulated as (10) or (13). SparsePOP [15] was used to generate SDPs of the test problems and then the ADAL method or SeDuMi was applied to the SDPs to compare the performance. The variable y in the ADAL method (3) was updated by applying the Cholesky factorization to $\mathcal{A}\mathcal{A}^*$, which was executed only once in the entire execution of the ADAL method. The primal variable X in (10) was initialized with the identity matrix I and the dual variable S in (13) with O. We also initialize $\mu = 1$, $\hat{x} = 1$ and $\hat{s} = 0$.

5.1 General polynomials

Numerical results on sparse polynomials, dense polynomials and a small illustrative example are discussed.

Sparse polynomials

If sparsity of the polynomial is exploited, the size of the positive semidefinite cones in SDPs becomes small. As a result, the computation of the search direction of the primal-dual interior point method and updating the matrix variable by (5) and (6) in the ADAL algorithm can be more efficient for sparse polynomials. We refer to [6, 15, 8] for details on sparsity of polynomials.

We experimented on test problems from Globallib [7], which were also used in SparsePOP [15]. Each test problem is a relaxation problem of the following polynomial optimization problem:

$$\min p(\boldsymbol{x}) \ \text{ subject to } \ \boldsymbol{x} \in B^n, \tag{27}$$

where B^n denotes the basic semialgebraic set $\{\boldsymbol{x} \in \mathbb{R}^n \mid p_i(\boldsymbol{x}) \geq 0 \ (i = 1, 2, ..., m)\}$. Note that the objective function of (27) has no constant term. Let p^* be the optimal value of the relaxation problem of POP (27). Then p^* is the lower bound of $p(\boldsymbol{x})$ over B^n. The function $p(\boldsymbol{x}) - q$ $(\boldsymbol{x} \in B^n)$ is nonnegative if and only if $q \leq p^*$.

Using the vector $u_d(\boldsymbol{x})$ in Lemma 2.3, the following problem is considered with $\gamma \geq 0$:

$$\text{Find } X \text{ s.t. } p(\boldsymbol{x}) - p^* + \gamma = u_d(\boldsymbol{x})^T X u_d(\boldsymbol{x}), \ X \succeq O, \ \boldsymbol{x} \in B^n.$$

The value of threshold "tol" for (8) in the ADAL algorithm was set to 10^{-4}, and the maximum number of iterations was set to 10,000.

Table 1 shows the results for two values of γ, 0 and $10^{12}\delta_0$. In the case of $\gamma = 0$, the size of the strictly feasible region δ_0 in Table 1 is the primal optimal value of SDP (24) obtained by SeDuMi. For $\gamma = 10^{12}\delta_0$, the corresponding size is shown in the column of δ_{12}. The number of iterations marked with $*$ in the column of ADAL indicates that the condition for the fast convergence in Theorem 3.1 or the condition for convergence in two iterations in Corollary 3.2 is satisfied. For the problem $p(\boldsymbol{x})$ with $\gamma = 0$, we see that the ADAL algorithm did not converge rapidly, taking many iterations. Extremely small strictly feasible region δ_0 and numerical difficulty caused by violating Assumption 2.1 may have caused the slow convergence. However, we observe in the column for $p(\boldsymbol{x}) + 10^{12}\delta_0$ that the sizes of the strictly feasible regions, δ_{12}, are larger than those in the column under δ_0, and the fast converge conditions are satisfied in many cases marked with $*$. As the minimum value of $p(\boldsymbol{x})$ is larger than 0, the strictly feasible region becomes larger, and the fast convergence is more likely to occur. This is the same results as in the case of trigonometric polynomials.

The ADAL method performs better than SeDuMi in terms of computational time and obtaining accurate solutions for the problems with large δ_{12}, say, greater than $1e$-01, in the column of $\gamma = 10^{12}\delta_0$. From the numbers of iterations required by SeDuMi in the column of $\gamma = 0$ and $\gamma = 10^{12}\delta_0$, we see that SeDuMi took less iterations for the problems of $\gamma = 10^{12}\delta_0$, but the differences in the iteration numbers between the problems with $\gamma = 0$ and $\gamma = 10^{12}\delta_0$ are greater for the ADAL method.

Table 1: Numerical results on sparse polynomials. "Itr" means the number of iterations and "Time(s)" CPU time in seconds. "> 10000" denotes that it takes more than 10000 iterations.

Problem	$\gamma = 0$					$\gamma = -10^{12}\delta_0$				
		SeDuMi		ADAL			SeDuMi		ADAL	
	δ_0	Itr	Time(s)	Itr	Time(s)	δ_{12}	Itr	Time(s)	Itr	Time(s)
Bex2_1_1	1.36e-11	19	2.17e+00	>10000	4.09e+01	1.48e-01	7	6.10e-01	30	2.26e-01
Bex2_1_2	1.37e-12	15	7.40e-01	1187	1.40e+00	5.64e-02	7	1.20e-01	40*	1.38e-01
Bex2_1_3	4.44e-12	14	1.25e+00	210	7.88e-01	6.56e-02	6	3.20e-01	34*	1.15e-01
Bex2_1_4	7.58e-13	12	5.80e-01	266	4.38e-01	2.19e-02	6	7.00e-02	85	1.88e-01
Bex2_1_5	2.85e-13	18	1.80e+00	1305	5.19e+00	1.40e-03	10	9.60e-01	3216*	1.04e+01
Bex2_1_8	2.86e-13	34	1.71e+01	2258	3.10e+01	2.95e-03	12	7.75e+00	2660	4.77e+01
Bex3_1_1	2.15e-12	19	1.85e+00	3871	2.58e+01	2.35e-02	7	4.50e-01	114	4.92e-01
Bex3_1_2	1.92e-12	17	7.90e-01	310	6.27e-01	5.62e-02	6	2.20e-01	44	1.24e-01
Bex3_1_4	1.23e-10	16	1.27e+00	3430	1.10e+01	2.23e+00	6	5.40e-01	10*	8.86e-02
Bex5_2_2_case1	9.69e-11	20	1.42e+00	>10000	2.82e+01	1.58e+00	6	4.90e-01	11*	9.62e-02
Bex5_2_2_case2	2.66e-11	20	1.80e+00	>10000	2.83e+01	4.39e-01	6	4.90e-01	17*	1.04e-01
Bex5_2_2_case3	3.41e-12	16	1.01e+00	>10000	2.61e+01	5.28e-02	7	5.10e-01	86	3.20e-01
Bex5_3_2	5.19e-09	30	1.03e+01	>10000	7.70e+01	4.17e+01	8	2.24e+00	9*	1.82e-01
Bex9_1_1	3.43e-13	20	1.74e+00	6009	1.33e+01	1.41e-02	8	4.50e-01	293	7.89e-01
Bex9_1_2	1.00e-09	34	5.34e+00	>10000	4.28e+01	1.31e+01	8	9.00e-01	11*	1.22e-01
Bex9_1_4	2.21e-09	18	1.10e+00	547	1.07e+00	3.87e+01	6	2.10e-01	7*	4.05e-02
Bex9_1_5	2.68e-14	20	1.21e+00	5823	9.51e+00	6.61e-04	10	3.50e-01	2617	4.50e+00
Bex9_1_8	1.45e-11	13	8.60e-01	901	1.78e+00	3.45e-01	6	2.10e-01	12*	5.24e-02
Bex9_2_1	7.54e-12	19	1.16e+00	1247	2.91e+00	1.91e-01	6	3.30e-01	26*	1.31e-01
Bex9_2_2	2.69e-09	23	1.56e+00	4857	5.23e+00	8.20e+01	5	1.30e-01	8*	2.64e-02
Bex9_2_3	2.67e-12	16	1.13e+00	2520	6.73e+00	6.67e-04	11	5.50e-01	4974	1.27e+01
Bex9_2_4	7.67e-14	18	7.70e-01	2472	3.89e+00	2.14e-03	9	3.00e-01	891*	1.37e+00
Bex9_2_5	1.94e-12	18	5.40e-01	1079	1.33e+00	7.60e-02	6	1.20e-01	40*	8.77e-02
Bex9_2_6	7.08e-13	21	2.57e+00	6516	2.47e+01	1.09e-02	8	7.90e-01	531	2.26e+00
Bex9_2_7	4.74e-10	18	1.46e+00	1445	3.59e+00	1.19e+01	6	3.50e-01	7*	4.20e-02
Bex9_2_8	1.81e-12	12	2.70e-01	322	2.23e-01	3.62e-01	4	8.00e-02	10*	1.66e-02
Balkyl	2.86e-14	24	4.52e+00	>10000	6.55e+01	1.74e-04	12	2.03e+00	>10000	6.51e+01
Bst_bpaf1a	1.31e-12	23	9.50e-01	4017	8.86e+00	2.85e-02	7	3.90e-01	95*	3.70e-01
Bst_bpaf1b	7.83e-13	18	8.70e-01	5352	1.13e+01	1.84e-02	7	3.30e-01	151	4.09e-01
Bst_e05	4.72e-12	16	5.90e-01	4966	3.36e+00	2.46e-01	5	1.20e-01	12*	2.97e-02
Bst_e07	1.12e-12	19	9.70e-01	3955	6.97e+00	1.81e-02	7	2.90e-01	137	3.33e-01
Bst_jcbpaf2	6.76e-13	18	1.67e+00	4497	1.45e+01	1.29e-02	8	7.70e-01	338	1.22e+00
Bhaverly	1.65e-11	23	1.29e+00	>10000	2.33e+01	2.64e-01	6	2.80e-01	20*	9.89e-02
alkylation	1.74e-13	16	1.88e+00	>10000	4.70e+01	1.29e-03	11	1.16e+00	3331	1.62e+01
Bst_robot	2.99e-11	15	7.60e-01	705	1.44e+00	5.15e-01	7	2.70e-01	17*	6.46e-02
st_cqpjk2	5.90e-11	12	3.00e-01	125	1.02e-01	2.96e+00	5	6.00e-02	2*	1.18e-02
st_e01	9.15e-11	18	4.60e-01	2167	8.32e-01	1.38e+01	6	1.10e-01	6*	1.36e-02
st_e09	1.05e-10	10	2.10e-01	1776	9.01e-01	1.08e+01	5	1.30e-01	7*	1.76e-02
st_e10	1.40e-09	11	1.40e-01	355	2.36e-01	2.76e+02	6	1.20e-01	6*	7.76e-03
st_e23	9.58e-12	12	2.10e-01	1528	6.33e-01	2.71e+00	4	8.00e-02	2*	5.86e-03
st_e34	1.28e-13	18	5.50e-01	771	1.29e+00	6.24e-03	8	1.90e-01	384*	6.51e-01
st_fp5	2.85e-13	18	1.76e+00	1305	4.31e+00	1.40e-03	10	9.20e-01	3216*	9.63e+00

Dense polynomials

We tested the SOS feasibility problem for the following dense polynomial:

$$p_\gamma(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^4 + 2 \sum_{1 \le i,j \le n} x_i^2 x_j^2 + \gamma, \quad \boldsymbol{x} \in \mathbb{R}^n, \ \gamma \ge 0. \tag{28}$$

The numerical results are displayed in Table 2. The experiments were conducted with parameters from $n = 5$ to 100, and $\gamma = 0$ and 1. The symbol 'OOM' indicates that SeDuMi failed to solve the problem due to out of memory. The value of threshold "tol" for (8) was set to 10^{-8}. The numbers marked with $*$ indicate the fast convergence of the ADAL algorithm. We observe that the problems with $\gamma = 1$ were solved very fast by the ADAL method, resulting in the fast convergence.

Table 2: Numerical results on the dense polynomial (28). $\binom{n+2}{2}$ denotes the size of positive semidefinite matrix and $\binom{n+4}{4}$ the number of constraints.

(n,γ)	$\binom{n+2}{2}$	$\binom{n+4}{4}$	SeDuMi		ADAL	
			Itr	Time(s)	Itr	Time(s)
(5,0)	21	126	22	1.3	41	0.4
(5,1)			5	0.2	9*	0.03
(10,0)	66	1,001	23	7.8	44	0.8
(10,1)			4	1.5	9*	0.03
(15,0)	136	3,876	24	317.6	65	2.7
(15,1)			4	46.2	10*	0.1
(20,0)	231	10,626	25	9736.7	73	9.5
(20,1)			4	1561.3	10*	0.4
(50,0)	1326	316,251	-	OOM	122	841.4
(50,1)			-	OOM	12*	45.9
(100,0)	5151	4,598,126	-	OOM	134	34656.0
(100,1)			-	OOM	14*	1510.3
(150,0)	11476	22,533,126	-	OOM	168	793100.0
(150,1)			-	OOM	16*	44476.4

The size of the SOS feasibility problem for (28) increases very rapidly since the size of positive semidefinite cone is $\binom{n+2}{2}$ and the number of constraints is $\binom{n+4}{4}$. The computational complexity of the primal-dual interior point method is known to be $O(\binom{n+2}{2}^6) = O(n^{12})$ and that of the ADAL method $O(\binom{n+2}{2}^3) = O(n^6)$. Thus, it takes a great deal of computation time for SeDuMi to solve the problems as n increases. For the ADAL method, the SOS feasibility problem of (28) is unconstrained and $\tilde{\mathcal{A}}\tilde{\mathcal{A}}^*$ is a diagonal matrix. As a result, the ADAL algorithm can update y without solving the linear equation. Consequently, the ADAL method can solve the problems much faster than SeDuMi, particularly in the case $n = 20$ and $\gamma = 1$.

For the largest instance ($n = 150$), we used a linux server with Opteron Processor 4386 (3.1 GHz) and 128 GB memory space. SeDuMi could not solve the problem with

$n \geq 50$ due to out of memory; for $n = 40$, SeDuMi required at least 130 GB memoy space. In contrast, the ADAL method consumed only 34 GB for the instance with $n = 150$ and $\gamma = 0$. In particular, the number of constraints, $m = 22,533,126$, in the SDP for $n = 150$ is twice larger than the largest problem solved by SDPNAL+ in [19]. For the instance with $n = 150$ and $\gamma = 1$, the ADAL method converged fast and an optimal solution with the desired accuracy could be obtained.

An illustrative example

We discuss that the convergence may differ depending on the formulations, in particular, between the primal and the dual in (10). Consider

$$p(\boldsymbol{x}) = x^4 - 2x^3 + 2x^2 - 2x + 1.$$

To represent as $p(\boldsymbol{x}) = u(\boldsymbol{x})^T X u(\boldsymbol{x})$ with $u(\boldsymbol{x}) = [1, x, x^2]$, we have the following constraints by equating the coefficients of $p(\boldsymbol{x})$ and $u(\boldsymbol{x})^T X u(\boldsymbol{x})$.

$$X_{1,1} = 1, X_{1,2} + X_{2,1} = -2,\ X_{1,3} + X_{2,2} + X_{3,1} = 2,\ X_{2,3} + X_{3,2} = -2,\ X_{3,3} = 1.$$

Let y_1 be a free variable for $X_{1,3} = X_{3,1}$ and an appropriate objective function be $-y_2$. Then the problem in dual form is

$$\begin{array}{ll} \min & -\begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \\ \text{s.t.} & \begin{pmatrix} 0 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 0 \end{pmatrix} y_1 + I y_2 + S = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}, \\ & S \succeq O. \end{array} \tag{29}$$

We note that the structure mentioned in Remark 2.4 cannot be found in this SDP.

Table 3 and Figures 1 and 2 illustrate the numerical results on (29). The horizontal axes in the two figures mean the number of iterations, and the vertical axes denote p_inf and d_inf (7) in Figures 1 and 2, respectively. The red lines are for (29) and the blue lines are for (10). The blue line in Figure 1 is not shown as p_inf = 0 was obtained throughout the iterations. The threshold "tol" in the termination condition (8) for the ADAL algorithm was set to 10^{-15}. We see that the convergence speed differs greatly between (29) and (10). For the formulation (29), the ADAL method gradually converged, requiring a large number of iterations. This is called the tail convergence. However, the ADAL method for (10) converged rapidly to a highly accurate solution and satisfied the condition for the convergence in Cororally 3.2.

5.2 Trigonometic polynomial

For the numerical tests on the SOS feasibility problem (13) for the univariate trigonometric polynomial $p_N(t)$ with the Dirichlet kernel, we used the sampling method described in (15). Then, a nonnegative polynomial $p_N(t)$ was generated by choosing the coefficients p_k, p_{-k} $(k = 1, 2, ..., N)$ from a uniform distribution $(-1, 1)$ and letting $p_0 = \sum_{k=1}^{N} (|p_k| +$

Table 3: Numerical results on (29). "Itr" means the number of iterations, "Time(s)" CPU time in seconds, "p_inf" and "d_inf" as defined in (7).

	SDP(10)	SDP(29)
Itr	2	158
Time(s)	0.1076	0.2300
p_inf	0	9.8210e-16
d_inf	5.5511e-17	1.1689e-16

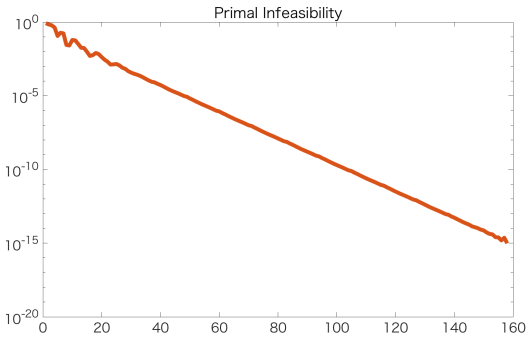

Figure 1: Primal infeasibility. — for (29).

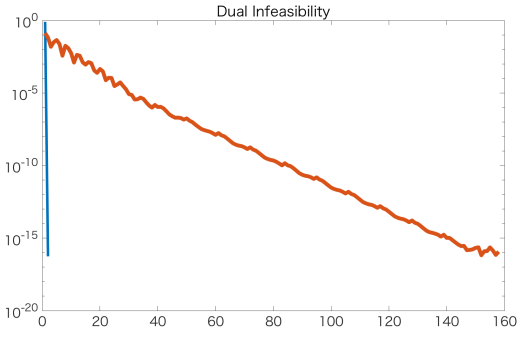

Figure 2: Dual infeasibility. — for (10) and — for (29).

$|p_{-k}|$). Let p_N^* be the minimum of $p_N(t)$ obtained by SparsePOP with SeDuMi. From Lemma 2.6, p_N^* can be obtained exactly by solving the SOS feasibility problem. The values of p_N^* were $46.9, 82.8, 164.4$ for $N = 50, 100, 200$, respectively. When we initialized X^0 with I, we obtained $19.9, 35.7, 84.0$ for the values of $\lambda_{min}(\breve{X})$ where $N = 50, 100, 200$, respectively.

The relationship between γ and the number of iterations is examined by solving the SOS feasibility problem for $p_N(t) + \gamma$ ($\gamma \leq 0$) with the ADAL method. We tested with $N = 50, 100, 200$. The value of γ is gradually decreased from 0 to $-p_N^*$ by 0.1 for $N = 50$, by 0.2 for $N = 100$, and by 1 for $N = 200$. The threshold "tol" in (8) was set to 10^{-8}. The experiment results are illustrated in Figures 3, 4, and 5. The horizontal axis represents γ, while the vertical axis indicates the number of iterations executed in the ADAL method.

In Figures 3, 4, and 5, we see that the ADAL method converges in two iterations for the values of $\gamma \in [-\lambda_{\min}(\breve{X}), 0]$, which confirms the result in Corollary 4.5. If γ becomes smaller than $-\lambda_{\min}(\breve{X})$, it takes more iterations than 2 iterations in Figure 5 where γ ranges from -80 to -150. In these cases, we observe that the condition for the fast convergence was still satisfied. We see that the number of iterations increases sharply when γ approaches $-p_N^*$. This is because the strictly feasible region of the SOS feasibility problem becomes very small as mentioned in Section 4.4. Hence, it is difficult to satisfy Assumption 2.1 numerically.

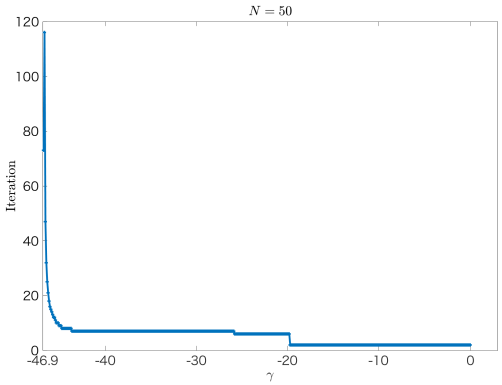

Figure 3: N=50 ($p_N^* = 46.9,\ \lambda_{\min}(\breve{X}) = 19.9$)

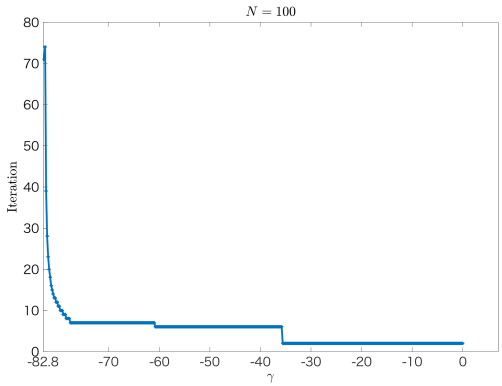

Figure 4: N=100 ($p_N^* = 82.8$, $\lambda_{\min}(\breve{X}) = 35.7$)

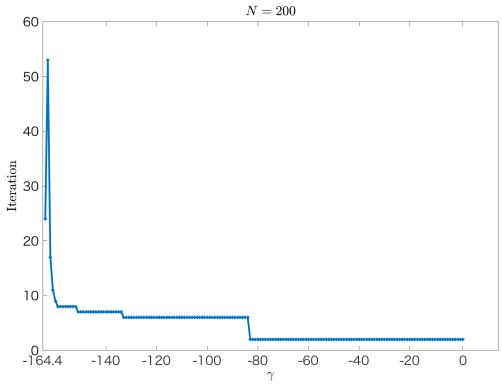

Figure 5: N=200 ($p_N^* = 164.4$, $\lambda_{\min}(\breve{X}) = 84.0$)

6 Concluding remarks

For the SOS feasibility problem of representing a (trigonometric) polynomial as an SOS, conditions for convergence of the ADAL method to an optimal solution in a few iterations have been proposed. The fast convergence of the ADAL method for the problem has been proved under the conditions. For trigonometric polynomials, we have proved that a given trigonometric polynomial can be adjusted to meet the conditions for the fast convergence of the ADAL method. The relation between the conditions and the size of the strictly feasible region of the SOS feasibility problem has been investigated.

The numerical results have been presented in Section 5 to numerically verify the fast convergence under the conditions. By enlarging the strictly feasible region of the SOS feasibility problem of general polynomials, we have demonstrated numerically that the conditions can be satisfied and the fast convergence of the ADAL method can be achieved. The theoretical results on the trigonometric polynomials have been also confirmed with the numerical experiments in Section 5.2.

The efficiency of the ADAL method over the primal-dual interior-point method as SeDuMi can be shown more clearly by testing on large-scale problems with an advanced computer with large memory than the machine used for the results in Section 5.

For many first-order methods, an initial value for X affects the overall convergence. In our experiments, the identity matrix I, a rough initialization, was used to initialize X. If the SOS feasibility problem of a polynomial cannot be solved fast, the ADAL method can be applied repeatedly by initializing X with the solution obtained in the previous application of the ADAL method. More precisely, we first add a positive number γ to the polynomial $p(\boldsymbol{x})$ and solve the SOS feasibility problem of $p(\boldsymbol{x}) + \gamma$ which is likely to satisfy the conditions for the fast convergence. Then use the obtained solution to initialize X for the next application of the ADAL method to the SOS feasibility problem of $p(\boldsymbol{x}) + \gamma'$ ($\gamma' < \gamma$). We plan to work on this repeated application of the ADAL method in the future.

References

[1] S. J. Benson and Y. Ye. Algorithm 875: DSDP5 – software for semidefinite programming. *ACM Trans. on Math. Softw.*, 34(3):16, 2008.

[2] G. Blekherman. There are significantly more nonnegative polynomials than sums of squares. *Israel J. Math.*, 153:355–380, 2006.

[3] L. Chua, D. Plaumann, R. Sinn, and C. Vinzant. Gram spectrahedra. arXiv:1608.00234v2, 2016.

[4] D. Cifuentes and P. A. Parrilo. Sampling algebraic varieties for sum of squares programs. *SIAM J. Optim.*, 27(4):2381–2404, 2017.

[5] B. Dumitrescu. Trigonometric polynomials positive on frequency domains and applications to 2-d fir filter design. *IEEE Trans. Singnal Processing*, 54(11):4282–4292, 2006.

[6] M. Fukuda, M. Kojima, K. Murota, and K. Nakata. Exploiting sparsity in semidefinite programming via matrix completion. I: General framework. *SIAM J. Optim.*, 11:647–674, 2000.

[7] GLOBAL Lib. GLOBAL library. `http://www.gamsworld.org/global/globallib.htm`. [Online; accessed 28-March-2018].

[8] S. Kim, M. Kojima, M. Mevissen, and M. Yamashita. Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion. *Math. Program.*, 129:33–68, 2011.

[9] J. Löfberg and P. A. Parrilo. From coefficients to samples: a new approach to sos optimization. In *The 43rd IEEE Conference on Decision and Control*, volume 3, pages 3154–3159. IEEE, 2004.

[10] P. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Math. Program.*, 96(2):293–320, 2003.

[11] H. Peyrl and P. A. Parrilo. Computing sum of squares decompositions with rational coefficients. *Theo. Comput. Sci.*, 409:269–281, 2008.

[12] T. Roh, B. Dumitrescu, and L. Vandenberghe. Multidimensional fir filter design via trigonometric sum-of-squares optimization. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):641–650, 2007.

[13] J. F. Sturm. SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods and Softw.*, 11&12:625–653, 1999.

[14] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Program.*, 95:189–217, 2003.

[15] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. Sums of squares and semidefinite programming relaxations for polynomial optimization problems with structured sparsity. *SIAM J. Optim.*, 17:218–242, 2006.

[16] Z. Wen, D. Goldfarb, and W. Yin. Alternating direction augmented lagrangian methods for semidefinite programming. *Math. Prog. Comp.*, 2(3):203–230, 2010.

[17] M. Yamashita, K. Fujisawa, M. Fukuda, K. Kobayashi, K. Nakata, and M. Nakata. Latest developments in the SDPA family for solving large-scale SDPs. In *Handbook on semidefinite, conic and polynomial optimization*, pages 687–713. Springer, 2012.

[18] M. Yamashita, K. Fujisawa, and M. Kojima. Implementation and evaluation of SDPA 6.0 (semidefinite programming algorithm 6.0). *Optim. Methods and Softw.*, 18(4):491–505, 2003.

[19] L. Q. Yang, D. F. Sun, and K. C. Toh. SDPNAL+: a majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Math. Prog. Comp.*, 7:331–366, 2015.

Appendix A: Proof for Lemma 4.3

Proof. Let $A_i = v(t_i)v(t_i)^T$. From (22), it holds that

$$\mathcal{A}(I) \;=\; (A_1 \bullet I, A_2 \bullet I, \ldots, A_{2N+1} \bullet I)^T = (||v(t_1)||^2, ||v(t_2)||^2, \ldots, ||v(t_{2N+1})||^2)^T = \boldsymbol{e}. \quad (30)$$

We show that $\mathcal{A}^*(\boldsymbol{e}) = \frac{2N+1}{N+1} I$.

$$\begin{aligned}(\mathcal{A}^*(\boldsymbol{e}))_{pq} &= \sum_{i=1}^{2N+1} (v_i v_i^T)_{pq} \\ &= \sum_{i=1}^{2N+1} K_{N+1}(t_i - (p-1)\tau) K_{N+1}(t_i - (q-1)\tau)\end{aligned}$$

For N, we consider two cases (i) $N+1 = 2m+1$ $(m \in \mathbb{N})$ and (ii) $N+1 = 2m+2$ $(m \in \mathbb{N})$.
(i) $N+1 = 2m+1$ $(m \in \mathbb{N})$:
Let $r = p-1$ and $s = q-1$. Then,

$$\begin{aligned}(\mathcal{A}^*(\boldsymbol{e}))_{pq} &= \frac{1}{(N+1)^2} \sum_{i=1}^{2N+1} \sum_{k_1=-m}^{m} \sum_{k_2=-m}^{m} \cos k_1(t_i - (p-1)\tau) \cos k_2(t_i - (q-1)\tau) \\ &= \frac{1}{2(N+1)^2} \sum_{k_1=-m}^{m} \sum_{k_2=-m}^{m} \sum_{i=1}^{2N+1} \left\{ \cos\left(-(k_1+k_2)\pi + \frac{(k_1+k_2)(i-1)2\pi}{2N+1} - (rk_1 + sk_2)\tau \right) \right. \\ &\quad \left. + \cos\left(-(k_1-k_2)\pi + \frac{(k_1-k_2)(i-1)2\pi}{2N+1} - (rk_1 - sk_2)\tau \right) \right\} \\ &= \frac{1}{2(N+1)^2} \sum_{k_1=-m}^{m} \sum_{k_2=-m}^{m} (-1)^{k_1+k_2} \sum_{i=1}^{2N+1} \left\{ \cos\left(\frac{(k_1+k_2)(i-1)2\pi}{2N+1} - (rk_1 + sk_2)\tau \right) \right. \\ &\quad \left. + \cos\left(\frac{(k_1-k_2)(i-1)2\pi}{2N+1} - (rk_1 - sk_2)\tau \right) \right\}.\end{aligned}$$

Let $\theta_i = \frac{(k_1+k_2)(i-1)2\pi}{2N+1}$ and $\phi_i = \frac{(k_1-k_2)(i-1)2\pi}{2N+1}$.

Let $\mathbb{Z}$ be the set of integers. Using Euler's formula, we have

$$\begin{aligned}\sum_{i=1}^{2N+1} \left\{ \cos(\theta_i - (rk_1 + sk_2)\tau) \right. &+ \left. \sqrt{-1} \sin(\theta_i - (rk_1 + sk_2)\tau) \right\} \quad (31) \\ &= e^{-\sqrt{-1}(rk_1+sk_2)\tau} \sum_{i=1}^{2N+1} e^{\sqrt{-1}\theta_i} \\ &= \begin{cases} (2N+1)e^{-\sqrt{-1}(p-q)k_1\tau} & \text{if} \quad k_1 = -k_2 \\ 0 & \text{if} \quad k_1 \neq -k_2 \end{cases}\end{aligned}$$

since the value of θ_i is 0 if $k_1 = -k_2$. If $k_1 \neq -k_2$, then $-2N \leq 2(k_1+k_2) \leq 2N$, $\frac{2(k_1+k_2)}{2N+1} \notin \mathbb{Z}$ and $e^{\sqrt{-1}k\pi} = 1$ $(\forall k \in 2\mathbb{N})$,

$$\sum_{i=1}^{2N+1} e^{\sqrt{-1}\theta_i} = \frac{1 - (e^{\sqrt{-1}\frac{2(k_1+k_2)\pi}{2N+1}})^{2N+1}}{1 - e^{\sqrt{-1}\frac{2(k_1+k_2)}{2N+1}\pi}} = 0.$$

Similarly,

$$\begin{aligned}\sum_{i=1}^{2N+1}\left\{\cos(\phi_i-(rk_1-sk_2)\tau) \;+\; \sqrt{-1}\sin(\phi_i-(rk_1-sk_2)\tau)\right\} \qquad (32)\\ = \; e^{-\sqrt{-1}(rk_1-sk_2)\tau}\sum_{i=1}^{2N+1} e^{\sqrt{-1}\phi_i}\\ = \; \begin{cases}(2N+1)e^{-\sqrt{-1}(p-q)k_1\tau} & \text{if} \quad k_1=-k_2\\ 0 & \text{if} \quad k_1\neq -k_2\end{cases}\end{aligned}$$

since $\phi=0$ if $k_1=k_2$. If $k_1\neq -k_2$, then $-2N\leq 2(k_1-k_2)\leq 2N$, $\frac{2(k_1-k_2)}{2N+1}\notin\mathbb{Z}$. As a result,

$$\sum_{i=1}^{2N+1} e^{\sqrt{-1}\phi_i}=\frac{1-(e^{\sqrt{-1}\frac{2(k_1-k_2)\pi}{2N+1}})^{2N+1}}{1-e^{\sqrt{-1}\frac{2(k_1-k_2)}{2N+1}\pi}}=0.$$

We see that

$$\begin{aligned}\sum_{i=1}^{2N+1}\cos(\theta_i-(rk_1+sk_2)\tau) &= \begin{cases}(2N+1)\cos(p-q)k_1\tau & \text{if} \quad k_1=-k_2\\ 0 & \text{if} \quad k_1\neq -k_2\end{cases}\\ \sum_{i=1}^{2N+1}\cos(\phi_i-(rk_1-sk_2)\tau) &= \begin{cases}(2N+1)\cos(p-q)k_1\tau & \text{if} \quad k_1=k_2\\ 0 & \text{if} \quad k_1\neq k_2.\end{cases}\end{aligned}$$

Consequently,

$$\begin{aligned}(\mathcal{A}^*(\boldsymbol{e}))_{pq} &= \frac{2N+1}{2(N+1)^2}\sum_{k_1=-m}^{m}\cos(p-q)k_1\tau\\ &= \frac{2N+1}{N+1}K_{N+1}\left(\frac{p-q}{N+1}2\pi\right)\\ &= \begin{cases}\frac{2N+1}{N+1} & \text{if} \quad p=q\\ 0 & \text{if} \quad p\neq q.\end{cases}\end{aligned}$$

(ii) $N+1=2m+2$ $(m\in\mathbb{N})$:

Similarly,

$$
\begin{aligned}
(\mathcal{A}^*(\boldsymbol{e}))_{pq} &= \left(\frac{2}{N+1}\right)^2 \sum_{k_1=0}^{m}\sum_{k_2=0}^{m}\sum_{i=1}^{2N+1} \cos\{(k_1+\frac{1}{2})(t_i-(p-1)\tau)\}\cos\{(k_2+\frac{1}{2})(t_i-(q-1)\tau)\} \\
&= \left(\frac{2}{N+1}\right)^2 \sum_{k_1=0}^{m}\sum_{k_2=0}^{m}\sum_{i=1}^{2N+1} \cos\{(k_1+\frac{1}{2})(t_i-r\tau)\}\cos\{(k_2+\frac{1}{2})(t_i-s\tau)\} \\
&= \frac{2}{(N+1)^2}\sum_{k_1=0}^{m}\sum_{k_2=0}^{m}\sum_{i=1}^{2N+1} \\
&\quad \left\{\cos\left((k_1+k_2+1)\pi+\frac{(k_1+k_2+1)(i-1)2\pi}{2N+1}-(k_1+\frac{1}{2})r-(k_2+\frac{1}{2})s\right)\right. \\
&\quad \left.+\cos\left((k_1-k_2)\pi+\frac{(k_1-k_2)(i-1)2\pi}{2N+1}-(k_1+\frac{1}{2})r+(k_2+\frac{1}{2})s\right)\right\} \\
&= \frac{2}{(N+1)^2}\sum_{k_1=0}^{m}\sum_{k_2=0}^{m}\sum_{i=1}^{2N+1} \\
&\quad \left\{(-1)^{k_1+k_2+1}\cos\left(\frac{(k_1+k_2+1)(i-1)2\pi}{2N+1}-(k_1+\frac{1}{2})r-(k_2+\frac{1}{2})s\right)\right. \\
&\quad \left.+(-1)^{k_1-k_2}\cos\left(\frac{(k_1-k_2)(i-1)2\pi}{2N+1}-(k_1+\frac{1}{2})r+(k_2+\frac{1}{2})s\right)\right\}.
\end{aligned}
$$

As in the case of $N+1=2m+1$,

$$
\begin{aligned}
\sum_{i=0}^{2N} e^{\sqrt{-1}\frac{k_1+k_2+1}{2N+1}2\pi i} &= 0, \\
\sum_{i=0}^{2N} e^{\sqrt{-1}\frac{k_1-k_2}{2N+1}2\pi i} &= \begin{cases} 2N+1 & \text{if} \quad k_1=k_2 \\ 0 & \text{if} \quad k_1\neq k_2 \end{cases}.
\end{aligned}
$$

From the definition of Dirichlet kernel,

$$
\begin{aligned}
(\mathcal{A}^*(\boldsymbol{e}))_{pq} = \frac{2}{(N+1)^2}\sum_{k_1=0}^{m}(2N+1)\cos\{(k_1+\frac{1}{2})(r-s)\tau\} &= \frac{2N+1}{N+1}K_{N+1}\big((r-s)\tau\big) \\
&= \frac{2N+1}{N+1}K_{N+1}\big((p-q)\tau\big) \\
&= \begin{cases} \frac{2N+1}{N+1} & \text{if} \quad p=q \\ 0 & \text{if} \quad p\neq q. \end{cases}
\end{aligned}
$$

We now obtained $\mathcal{A}^*(\boldsymbol{e})=\frac{2N+1}{N+1}I$. Thus, by (30), we obtain

$$
(\mathcal{A}\mathcal{A}^*)^{-1}(\boldsymbol{e}) = \left(\frac{2N+1}{N+1}\right)^{-1}\boldsymbol{e} \quad \text{and} \quad \mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}(\boldsymbol{e})\big) = I.
$$

Substituting (30), we derive $\mathcal{A}^*\big((\mathcal{A}\mathcal{A}^*)^{-1}\mathcal{A}(I)\big)=I$.

□